IMPLEMENTASI PENGAMANAN BASIS DATA DENGAN TEKNIK ENKRIPSI KRIPTOGRAFI SIMETRIS

Zil Fadli[1], Taslim [2]
[1,2] Teknik Informatika, Fakultas Ilmu Komputer
Univerwsitas Lancang Kuning Pekanbaru
Jl. Yos Sudarso, Rumbai – Pekanbaru
Email: zil_fadli@gmail.com, sri_paduka@yahoo.com

ABSTRAK

Salah satu aspek penting dari pengolahan data adalah masalah keamanan data dan kerahasiaan data. Data - data yang terdapat dalam suatu sistem pengolahan data diharapkan untuk mengamankan pihak yang tidak berhak tidak dapat mengaksesnya. Untuk bentuk dikembangkan keamanan dan kerahasiaan data dengan melakukan menu password pengenkripsian dengan menggunakan DES yang diharapkan tingkat keamanan data terjamin.

Kata kunci: data, keamanan, kerahasiaan, DES, password,

ABSTRACT

One important aspect of data processing is the problem of data security and confidentiality of data. Data - the data contained in a data processing system is expected to secure unauthorized parties can not access them. For the developed form of security and confidentiality of data by performing pengenkripsian password menu by using the DES which is expected to be the level of data security is guaranteed.

Key word: *Data, security,confidentiality, DES, password,*

1. PENDAHULUAN

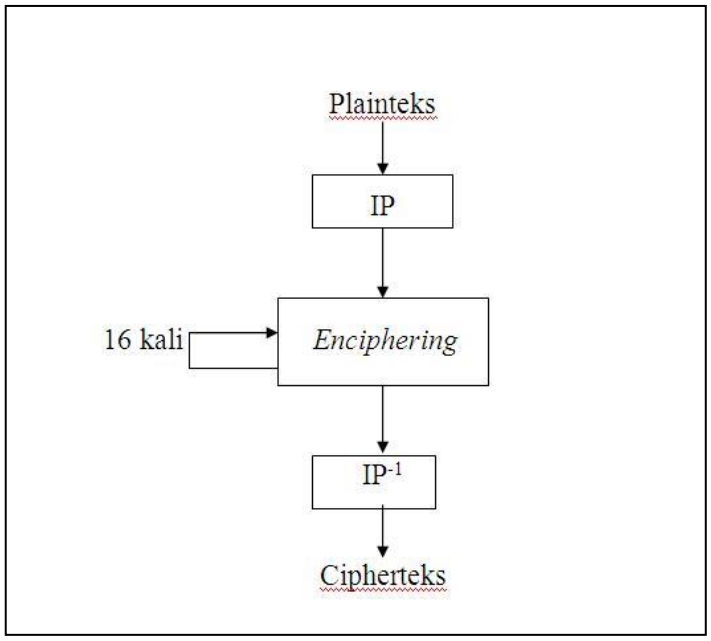
Kriptografi dapat digunakan untuk mengamankan data. Oleh karena itu, pengguna basis data membutuhkan bantuan untuk memenuhi kebutuhan keamanan akan data yang disimpannya. Penerapan kriptografi ini akan difokuskan bagaimana kriptografi dapat mengamankan data sampai pada level baris (*row*) dan kolom (*field*) dengan tetap memperhatikan integritas data dan kewenangan setiap pengguna basis data. Algoritma kriptografi yang akan digunakan ialah algoritma kriptografi simetris dan bersifat *stream cipher* sehingga data hasil enkripsi (cipherteks) mempunyai ukuran yang sama dengan data asli (plainteks). Teknik kriptografi simetris dipilih karena diharapkan dengan

algoritma ini proses enkripsi – dekripsi data dapat dilakukan dengan waktu yang lebih cepat dibandingkan dengan algoritma kriptografi kunci publik (asimetris).

2. DASAR TEORI

2.1 Data Encryption Standard (DES)

DES termasuk ke dalam sistem kriptografi simetri dan tergolong jenis *cipher* blok. DES beroperasi pada ukuran blok 64 bit. DES mengenkripsikan 64 bit plainteks menjadi 64 bit cipherteks dengan menggunakan 56 bit kunci internal (*internal key*) atau upa-kunci (*subkey*). Kunci internal dibangkitkan dari kunci eksternal (*external key*) yang panjangnya 64 bit. secara global algorima digambarkan sebagai berikut :



Gambar 2.1 Algoritma *Data Encryption Standard* (DES)

Keterangan dari gambar diatas adalah :

1. Blok plainteks dipermutasi dengan matriks permutasi awal (*initial permutation* atau IP).
2. Hasil permutasi awal kemudian di-*enciphering*- sebanyak 16 kali (16 putaran). Setiap putaran menggunakan kunci internal yang berbeda.
3. Hasil enciphering kemudian dipermutasi dengan matriks permutasi balikan (*invers initial permutation* atau IP^{-1}) menjadi blok cipherteks.

Plantext adalah text mentah. misal huruf a nilai binarynya 01100001. jadi kalau misal ada plaintext: bayu, maka binarynya 01100010011000010111100101110101. huruf besar dan huruf kecil berbeda binarynya. setelah itu binary yang tersusun dari plaintext dipecah-pecah tiap 64 bit. nanti tiap 64 bit itu akan dipermutasikan sama matriks permutasi (IP). berikut matriks IP

Tabel 2.1 Matriks permutasi *Initial Permutation* (IP)

58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6	64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1	59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5	63	55	47	39	31	23	15	7

cara baca matriks IP ini adalah dari kiri atas ke kanan. arti dari matriks ini adalah memindahkan/mengacak posisi 64 bit plaintext. cara kerjanya adalah dengan memindahkan bit ke 58 ke posisi 1, bit ke 50 ke posisi 2, dst. contohnya seperti ini:
plaintext : bayucaem
binary : 0110001001100001011110010111010101100011011000010110010101101101
jika sudah diacak oleh matriks IP akan menjadi:

1	1	1	1	1	1	1	1	0	0	0	0	1	1	0	0
1	1	0	0	1	0	0	0	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1

setelah tahap permutasi selesai dilakukan maka sekarang menuju ke tahap enchipering. disni butuh yang namanya kunci eksternal yang akan membentuk kunci internal. jadi gambarannya sang user jika ingin mengenkripsi maka ada dua yang harus diinputkan yaitu plaintextnya dan kuncinya sepanjang 16 digit hexadesimal. kunci ini juga (selanjutnya disebut kunci eksternal) akan digunakan pada saat mendekripsi DES.
diperlukan 16 kunci internal untuk digunakan dalam putaran enchipering. untuk mendapatkannya pertama, kunci external yang diinputkan oleh user dalam bentuk hexadesimal diubah kebentuk biner kemudian dipermutasikan oleh matriks PC-1. ini matriksnya :

Tabel 2.2 Matriks permutasi *Chiper-1* (PC-1)

57	49	41	33	25	17	9	1	58	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

cara permutasinya sama dengan permutasi plaintext dan mengahsilkan 56 bit dari 64 bit kunci eksternal. setelah didapatkan hasil dari permutasi maka hasil permutasi tadi dibagi dua, yaitu bagian kiri dan kanan masing-masing 28 bit. Selanjutnya, kedua bagian digeser ke kiri (*left shift*) sepanjang satu atau dua bit bergantung pada tiap putaran. Operasi pergeseran bersifat *wrapping* atau *round-shift*. Jumlah pergeseran pada setiap putaran ditunjukkan pada Tabel 2.3 sbb:

Tabel 2.3 Jumlah pergeseran pada setiap putaran

Putaran, *i*	Jumlah pergeseran bit
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

setelah digeser, masing-masing digabungkan kembali dan kembali lagi dipermutasikan. tetapi matriks yang digunakan adalah matriks pc-2 :

Tabel 2.4 Matriks permutasi *Chiper-2* (PC-2)

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

ilustrasinya begini:
key eksternal inputan user : ab12781bac253987
binernya:

1 0 1 0 1 0 1 1 0 0 0 1 0 0 1 0
0 1 1 1 1 0 0 0 0 0 0 1 1 0 1 1
1 0 1 0 1 1 0 0 0 0 1 0 0 1 0 1
0 0 1 1 1 0 0 1 1 0 0 0 0 1 1 1

setelah dipermutasikan dengan pc-1 dan dibagi menjadi dua bagian (L0 dan R0)

L0

1 0 0 1 0 0 0 1 0 0 0 0 0 1
0 0 0 1 1 1 0 1 0 1 0 1 0 0

R0

1 0 0 0 1 0 1 1 1 0 1 1 0 0
0 0 0 1 0 1 1 1 0 1 1 1 1 0

karena ini adalah putaran pertama maka L0 dan R0 masing-masing digeser kekiri 1 kali menurut tabel pergeseran diatas, hasilnya begini setelah L0 dan R0 digeser dan digabung kembali kemudian dipermutasikan dengan pc-2

0 1 0 0 0 0 1 1 0 0 1 0
1 1 0 0 0 0 0 1 0 0 1 0
0 1 0 0 1 0 0 0 1 1 0 1
0 1 1 1 1 1 0 0 0 1 0 1

dalam hexadecimal : 432c1248d7c5
itulah key internal pertama yang didapatkan. untuk mendapatkan key/kunci internal yang kedua cukup key yang pertama dibagi masing-masing 24 bit kiri dan kanan setelah itu digeser menurut table pergesaran dan dipermutasikan dengan pc-2. berikut key internal lengkapnya :

Tabel 2.5 Hasil *Enchipering*

Key eksternal : ab12781bac253987	
No	key
1	432c1248d7c5
2	50422d7cb8e8
3	81912420fc7f
4	800ae72fbcb2
5	b17220ad4d73
6	8017e00fca56
7	d05255d5c5d4
8	05d3408986cd
9	e621187136f8
10	0e861139b82f
11	4f181a267cb6
12	2ea0c82d29f7
13	1a4c0aa7c8d3
14	682918478757
15	04ac0d9f85cc
16	9805c455366e

selesai sudah tahap dari enchipering DES, hasil dari enchipering eksternal key berupa hexadesimal diubah menjadi bilangan biner dan keluaran dari tahap enchipering ini akan di invers. caranya yaitu dilakukan permutasi oleh matriks IP^{-1}

Tabel 2.6 . ***Invers Initial Permutation*** **(IP^{-1})**

40	8	48	16	56	24	64	32	39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30	37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28	35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26	33	1	41	9	49	17	57	25

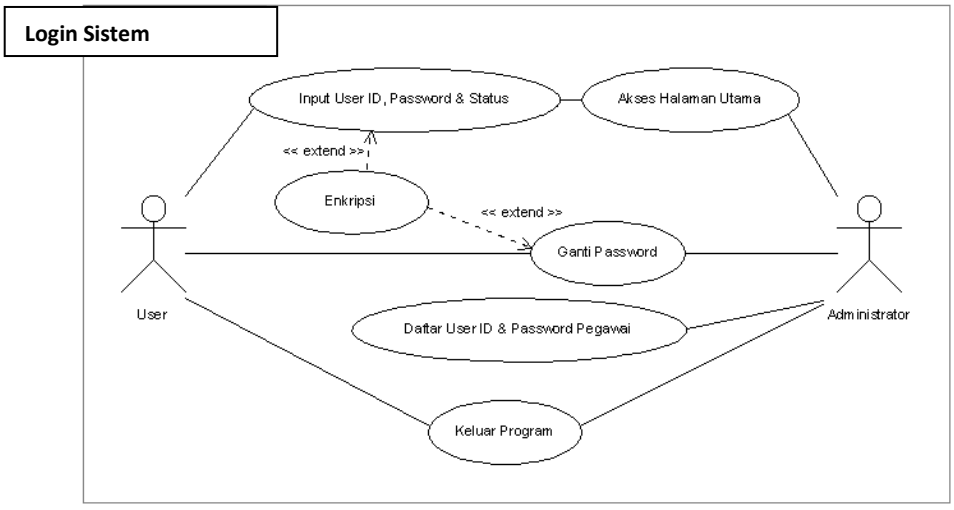
cara permutasinya sama dengan permutasi plaintext dan mengahsilkan 56 bit dari 64 dan setelah itu maka dihasilkan chiper text yang merupakan hasil penyandian dari plaintext.

3. RANCANGAN SISTEM

Dalam sistem baru ini, metode Perancangan yang diusulkan ini menggunakan UML(*Unifed Modeling Language*) dalam perancangannya .Berikut adalah diagram-diagram yang digunakan dalam perancangan tersebut.

1. *Use Case* Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use Case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.



Gambar 3.1 ***Use Case*** **diagram aplikasi enkripsi login database**

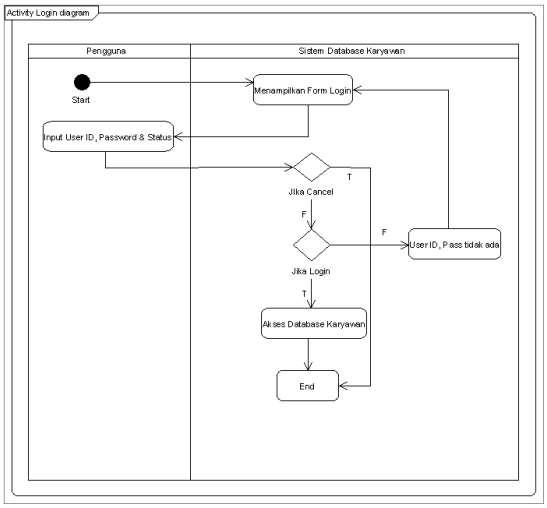
Use Case Login Database mempunyai langkah – langkah:

1. User dan Administrator diharuskan Menginput User ID, Password dan Status terlebih dahulu
2. Setelah menginput User ID, Password dan Status maka Password akan dienkripsi.
3. Stelah langkah kedua selesai barulah User dan Administrator dapat mengakses database
4. Administrator memiliki akses untuk mendaftarkan user baru.

2. *Activity Diagram*

Activity diagram\ menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekus. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

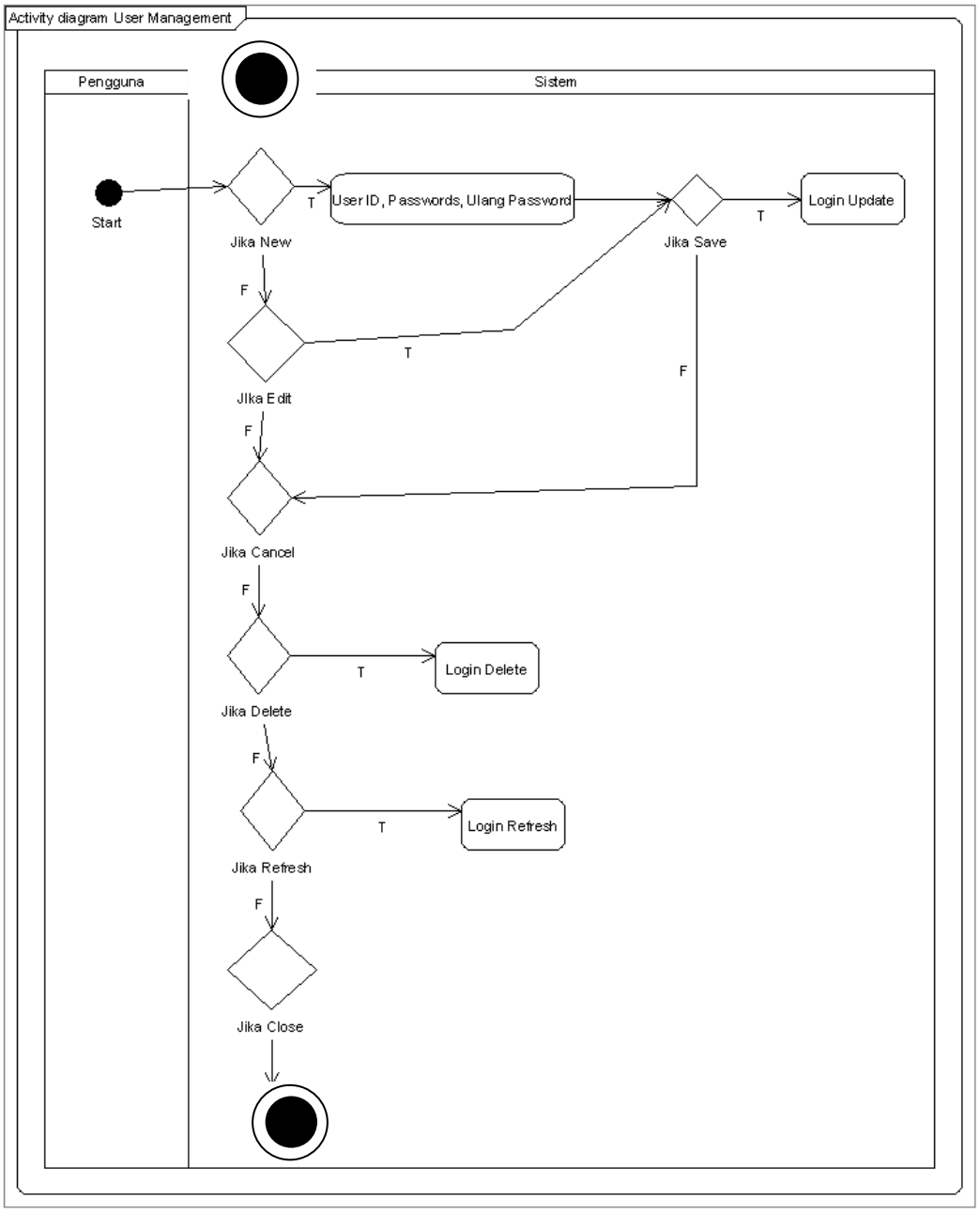
Diagram Activity dapat dibagi menjadi beberapa jalur kelompok yang menunjukkan obyek yang mana yang bertanggung jawab untuk suatu aktifitas. Peralihan tunggal (*single transition*) timbul dari setiap adanya *activity* (aktifitas), yang saling menghubungi pada aktifitas berikutnya. Sebuah *transition* (transisi) dapat membuat cabang ke dua atau lebih percabangan *exclusive transition* (transisi eksklusif). Label *Guard Expression* (ada di dalam [ ]) yang menerangkan output (keluaran) dari percabangan. percabangan akan menghasilkan bentuk menyerupai bentuk intan. *transition* bisa bercabang menjadi beberapa aktifitas paralel yang disebut **Fork.** *Fork* beserta *join* (gabungan dari hasil output *fork*) dalam diagram berbentuk *solid bar* (batang penuh). Berikut *Activity* Login *Diagram* dari aplikasi enkripsi login database :



Gambar 3.2 *Activity login* diagram aplikasi enkripsi login database

Diagram aktivitas menggambarkan berbagai alir aktivitas pada suatu sistem perangkat lunak, bagaimana masing-masing alir berawal, keputusan yang mungkin

terjadi, dan bagaimana alir berakhir. Diagram aktivitas loginpada Gambar 3.3 menggambarkan alir aktivitas pengguna mulai dari proses input user id, input password dan status, masuk ke menu utama, dan menutup aplikasi.

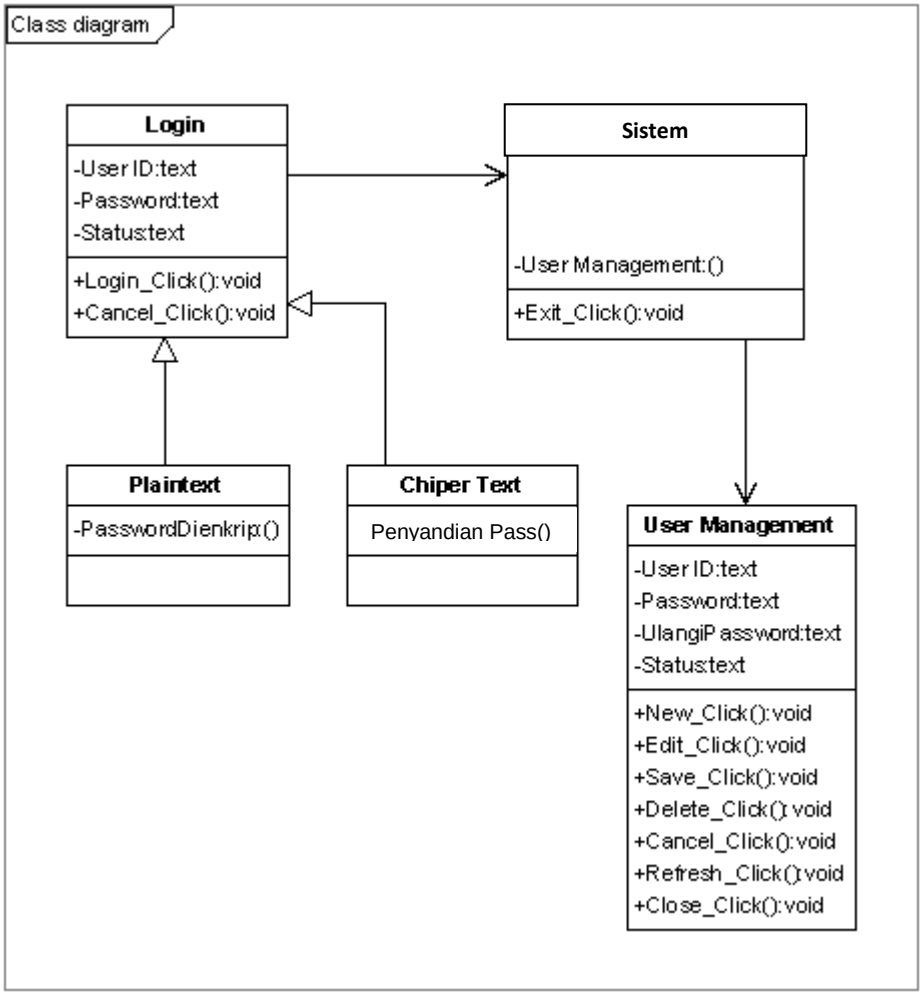


Gambar 3.3 *Activity login* diagram aplikasi enkripsi login database

3. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class*

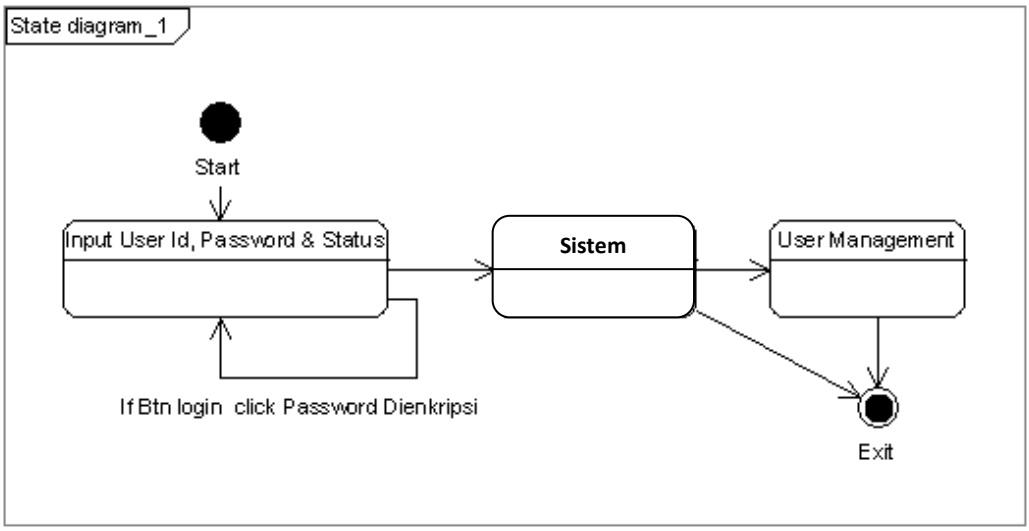
menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. Berikut *Class Diagram* dari aplikasi enkripsi login database :



Gambar 3.4 *Class diagram* aplikasi enkripsi login database

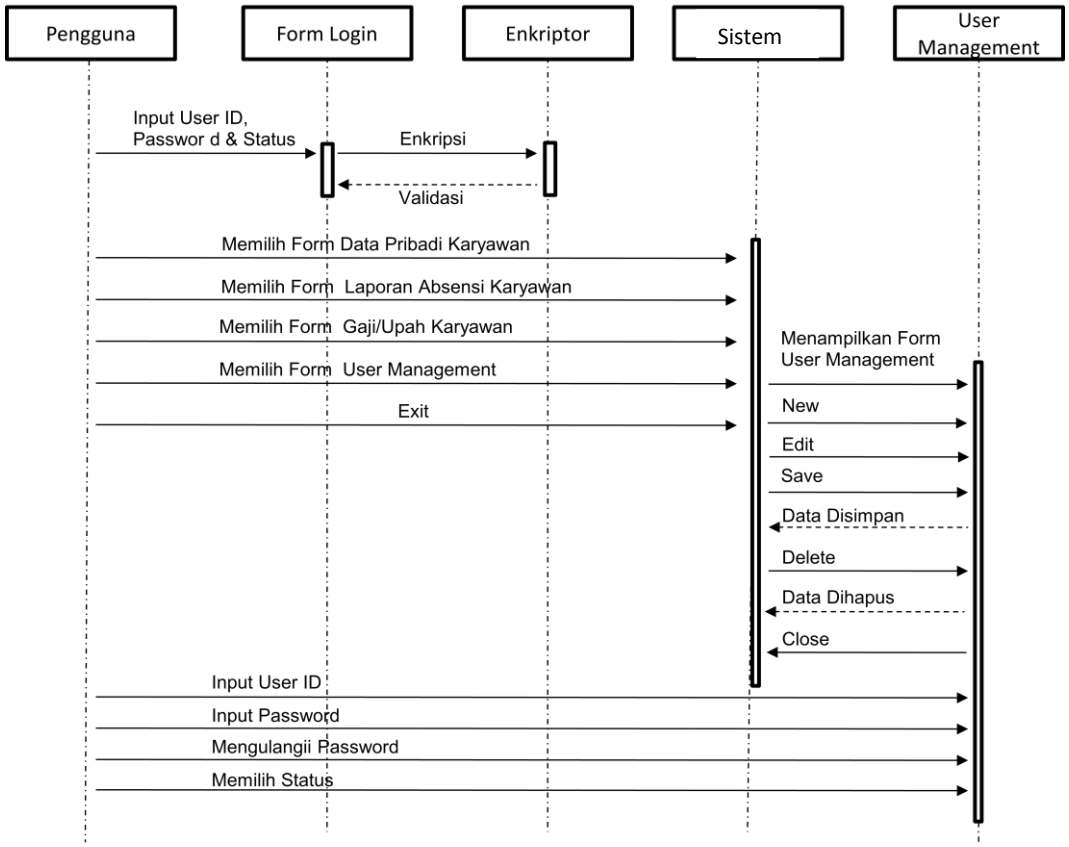
4. Statechart Diagram

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*). Berikut *Class Diagram* dari aplikasi enkripsi login database :



Gambar 3.5 *Statechart diagram* aplikasi enkripsi login database *Sequence Diagram*

Diagram sequence merupakan salah satu diagram Interaction yang menjelaskan bagaimana suatu operasi itu dilakukan; *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Obyek-obyek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut. *Lifeline* adalah garis dot (putus-putus) vertikal menerangkan waktu terjadinya suatu obyek. Setiap panah yang ada adalah pemanggilan suatu pesan. Panah berasal dari pengirim ke bagian paling atas dari batang kegiatan (*activation bar*) dari suatu pesan pada *lifeline* penerima. *Activation bar* menerangkan lamanya suatu pesan diproses.
Berikut *Sequence Diagram* dari aplikasi enkripsi login database :



Gambar 3.6 *Sequence diagram* aplikasi enkripsi login database

2. **Perancangan input**

Adapun disini penulis hanya menampilkan bentuk struktur data tampilan rancangan basis data login.

Data Login

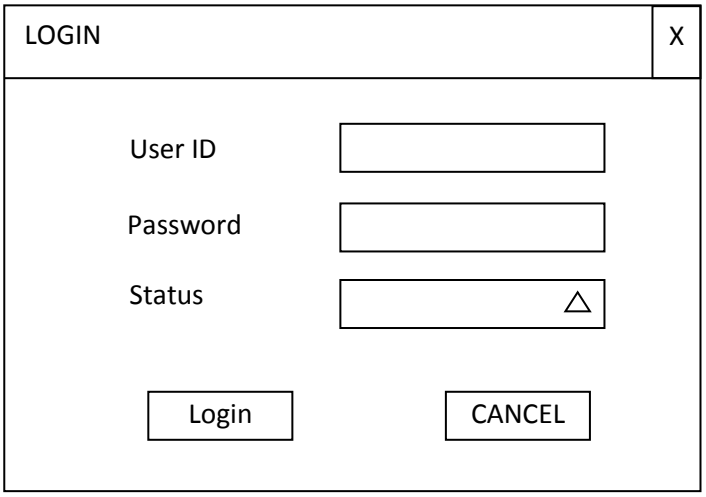
Tabel 3.1. Data login

Field Name	Type Field	Width	Keterangan
UserId	Text	8	Untuk menyimpan user name pengguna
Pass	Text	10	Untuk menyimpan password pengguna
Status	Text	15	Untuk menyimpan status pengguna

3. **Perancangan Form output**

Perancangan program enkripsi ini digunakan untuk membantu pengamanan database login pada program sistem informasi

Perancangan Form



Gambar 3.7 Perancangan Form

Sebelum masuk kedalam program terdapat menu yang pertama yaitu menu login. Jadi setiap pengguna yang akan menggunakan program sistem informasi ini harus melakukan login. Dalam menu login ini pengguna harus mengisikan user id dan password.

4. KESIMPULAN

Berdasarkan hasil pengamatan dan pengujian yang telah dilakukan maka dapat disimpulkan bahwa dalam penelitian ini :

1. Teknik Enkripsi Kriptografi Simetris dapat di implementasikan untuk pengamanan basis data.
2. Proses validasi dan Digital Signature yang ditambahkan pada perangkat lunak dapat berjalan dengan baik dan dapat mendeteksi adanya perubahan pada teks yang dikirm serta dapat menjamin keutuhan dan keaslian data dari pengirim informasi.

DAFTAR PUSTAKA

Novi Dian Nathasia, Anang Eko Wicaksono. 2011 .*Penerapan Teknik Kriptografi Stream Chiper Untuk Pengamanan Basis Data*. Jurnal Basis Data, ICT Research Center UNAS.

Munawar. 2012. *Perancangan Algorithma Sistem Keamanan Data Menggunakan Metoda Kriptografi Asimetris.* Jurnal Komputer dan Informatika.

Semuil Tjiharjadi, Marvin Chandra Wijaya. 2009. *Pengamanan Data Menggunakan Metoda Enkripsi Simetridengan Algoritma Feal*. Seminar Nasional Aplikasi Teknologi Informasi .